Intel® Xeon™ Processor MP

Specification Update

April 2005

Notice: The Intel® Xeon™ Processor MP and Intel® Xeon™ Processor MP with up to 4-MB cache on 0.13-micron process may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are documented in this Specification Update.

Document Number: 290741-034

Contents

Revision History

Version	Description	Date
-001	• Initial release.	March 2002
-002	• Addition of “Mixed Steppings in MP Systems” section. • Added five new Documentation Changes. • Added erratum O39.	April 2002
-003	• Added PWRGOOD Specification Change. • Added errata O40. • Updated errata O12, O29. • Added Specification Changes O1 and O2. • Added Documentation Changes O1-O3.	May 2002
-004	• Added errata O41 and O42. • Added new Documentation Changes O1- O2	June 2002
-005	• Added new erratum O43 • Added new Documentation Changes O3 - O12.	July 2002
-006	• Added new erratum O44.	August 2002
-007	• Added new errata 045 • Edited erratum O13. • Added new Documentation Changes O3 - O24.	September 2002
-008	• Removed erratum previously numbered O45 as not applicable. • Added new erratum O45. • Edited erratum O32. • Added new Doc Changes O25 - O32.	October 2002
-009	• Added new processor - Intel® Xeon™ Processor MP with up to 2-MB L3 cache on 0.13-micron process. • Added new processor S-Specs. • Added new Processor Signature (0F22h). • Added new identification information. • Added erratum O46 - O51. • Added Specification Clarification O1. • Added Specification Change O1. • Updated Mixed Stepping Matrix.	November 2002
-010	• Added Documentation Changes notice.	December 2002
-011	• Added new errata O52 - O53. • Deleted old Documentation Changes.	February 2003
-012	• Added new errata O54 - O55.	March 2003
-013	• No New updates.	April 2003
-014	• Added Specification Clarification O1 - O2.	May 2003
-015	• Added erratum O56. • Updated erratum O46.	June 2003

Version	Description	Date
-016	• Added B0 Stepping. • Added new processor S-Specs. • Added new Processor Signature (0F25h). • Added new identification information. • Added erratum O57. • Updated Summary of Errata Table. • Updated Table 2. • Removed Specification Clarifications and Specification Changes.	July 2003
-017	• Added erratum O58.	July 2003
-018	• Updated incorrect stepping information (Updated B0 to B1 stepping).	July 2003
-019	• Added errata O59 - O60. • Updated erratum O58. • Added Specification Change O1.	August 2003
-020	• Added erratum O60. • Updated erratum O10, O20.	September 2003
-021	• Added errata O61 - O62.	October 2003
-022	• Added erratum O63.	November 2003
-023	• Added errata O64 and O65. • Added C0 Stepping. • Added new processor S-Specs. • Added new Processor Signature (0F26h). • Added new identification information. • Updated Summary of Errata Table. • Updated Table 2.	March 2004
-024	• Updated errata O22 and O63.	April 2004
-025	• Updated errata O11 and O48. • Added erratum O66.	May 2004
-026	• Added errata O67, O68, O69.	June 2004
-027	• Updated affected documents listed under Preface, Specification Changes, Specification Clarifications, and Documentations Changes.	July 2004
-028	• Added erratum O70. • Updated erratum O52.	August 2004
-029	• Added errata O71-O72.	September 2004
-030	• Added erratum O73.	September 2004
-031	• Added errata O74-O76.	October 2004
-032	• Added erratum O77.	November 2004
-033	• Added erratum O78.	December 2004
-034	• Added Specification Clarification O1.	April 2005

Preface

Affected/Related Documents

This document is an update to the specifications contained in the following documents:

Document Title	Document Number
Intel® Xeon™ Processor MP Datasheet	290740
Intel® Xeon™ Processor MP with up to 4MB L3 Cache (on 0.13 Micron Process) Datasheet	251931
IA-32 Intel® Architecture Software Developer's Manual, Volumes 1, 2A, 2B and 3	253665, 253666, 253667 and 253668, respectively
Intel® Extended Memory 64 Technology Software Developer's Guide, Volume 1	300834
Intel® Extended Memory 64 Technology Software Developer's Guide, Volume 2	300835

It is intended for hardware system manufacturers and software developers of applications, operating systems, or tools. It contains Errata, Documentation Changes, Specification Clarifications and Specification Changes.

Nomenclature

S-Spec Number is a five-digit code used to identify products. Products are differentiated by their unique characteristics, e.g., core speed, L2 cache size, package type, etc. as described in the processor identification information table. Care should be taken to read all notes associated with each S-Spec number.

Errata are design defects or errors. Errata may cause the behavior of the Intel® Xeon™ Processor MP and Intel® Xeon™ Processor MP with up to 4-MB L3 cache on 0.13- micron- micron process to deviate from published specifications. Hardware and software designed to be used with any given processor must assume that all errata documented for that processor are present on all devices unless otherwise noted.

Documentation Changes include typos, errors, or omissions from the current published specifications. These changes will be incorporated in the next release of the specifications.

Specification Clarifications describe a specification in greater detail or further highlight a specification's impact to a complex design situation. These clarifications will be incorporated in the next release of the specifications.

Specification Changes are modifications to the current published specifications. These changes will be incorporated in the next release of the specifications.

Identification Information

Intel® Xeon™ Processor MP and Intel® Xeon™ Processor MP with up to 4MB L3 Cache (on 0.13 Micron Process) Markings (603-pin INT-mPGA)

Figure 1. Top Side Processor Marking – Production Part

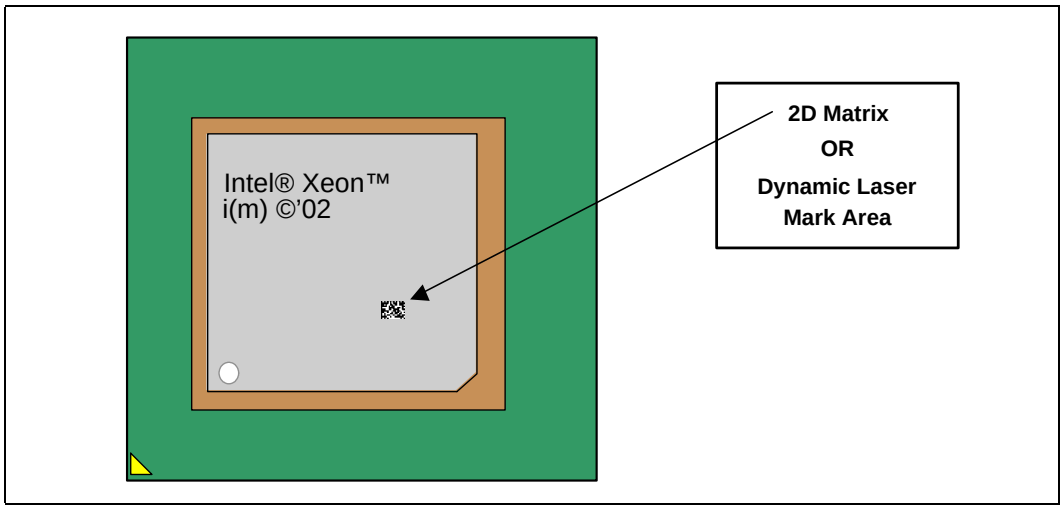
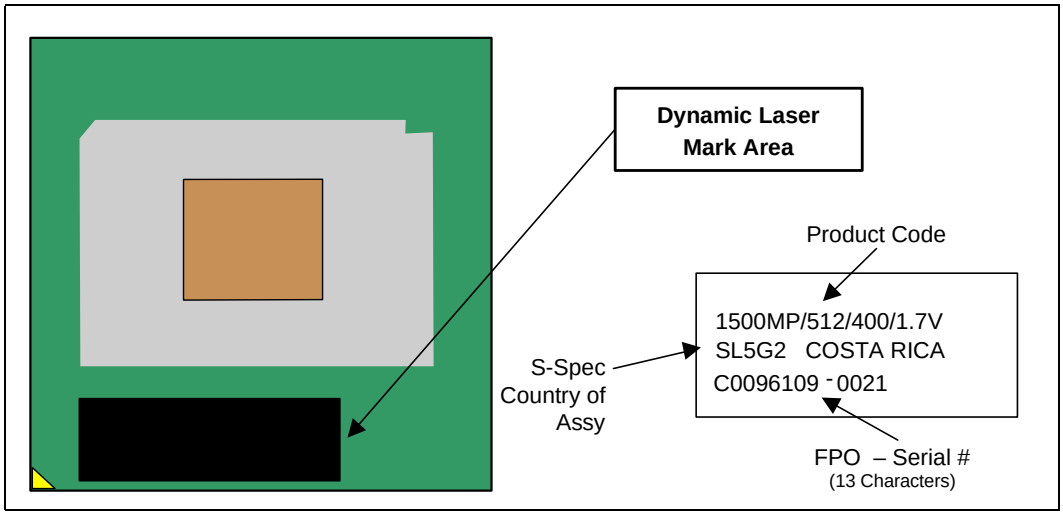


Figure 2. Bottom Side Processor Marking



The Intel Xeon processor MP can be identified by the following values:

Family[1]	Model[2]	Brand ID[3]
1111	0001	00001110
1111	0010	00001011[4]

1. The Family corresponds to bits [11:8] of the EDX register after RESET, bits [11:8] of the EAX register after the CPUID instruction is executed with a 1 in the EAX register, and the generation field of the Device ID register accessible through Boundary Scan.
2. The Model corresponds to bits [7:4] of the EDX register after RESET, bits [7:4] of the EAX register after the CPUID instruction is executed with a 1 in the EAX register, and the model field of the Device ID register accessible through Boundary Scan.
3. The Brand ID corresponds to bits [7:0] of the EBX register after the CPUID instruction is executed with a 1 in the EAX register.
4. Brand ID and Model for Intel® Xeon™ Processor MP with up to 4-MB L3 cache on 0.13- micron process.

Cache and TLB descriptor parameters are provided in the EAX, EBX, ECX and EDX registers after the CPUID instruction is executed with a 2 in the EAX register. Please refer to the *Intel Processor Identification and the CPUID Instruction Application Note* (AP-485) for further information on the CPUID instruction.

Table 1. Intel® Xeon™ Processor MP Identification Information

S-Spec	Core Stepping	Processor Signature	Speed Core/Data Bus (GHz/MHz)	L2 Cache Size	L3 Cache Size	Hyper-Threading Technology	Processor Interposer Revision	Package And Revision[2]	S-Spec Notes
SL5G8	C0	0F11h	1.60/400	256KB	1- MB	Yes	B1	42.5 mm OLGA rev 1.0	1
SL5S4	C0	0F11h	1.60/400	256KB	1- MB	Yes	B1	42.5 mm OLGA rev 1.0	1, 3
SL5FZ	C0	0F11h	1.40/400	256KB	512-KB	Yes	B1	42.5 mm OLGA rev 1.0	1
SL5RV	C0	0F11h	1.40/400	256KB	512-KB	Yes	B1	42.5 mm OLGA rev 1.0	1, 3
SL5G2	C0	0F11h	1.50/400	256KB	512-KB	Yes	B1	42.5 mm OLGA rev 1.0	1
SL5RW	C0	0F11h	1.50/400	256KB	512-KB	Yes	B1	42.5 mm OLGA rev 1.0	1,3
SL6GZ SL6KB	A0	0F22h	1.50/400	512-KB	1-MB	Yes	01	603-pin micro-PGA interposer with 42.5 mm FC-BGA package	1, 2,4 1, 2, 3, 4
SL6H2 SL6KC	A0	0F22h	1.90/400	512-KB	1-MB	Yes	01	603-pin micro-PGA interposer with 42.5 mm FC-BGA package	1, 2, 4 1, 2, 3, 4
SL66Z SL6KD	A0	0F22h	2/400	512-KB	2-MB	Yes	01	603-pin micro-PGA interposer with 42.5 mm FC-BGA package	1, 2, 4 1, 2, 3, 4
SL6YJ SL6Z6	B1	0F25h	2/400	512-KB	1-MB	Yes	01	603-pin micro-PGA interposer with 42.5 mm FC-BGA package	1, 2, 4 1, 2, 3, 4

Table 1. Intel® Xeon™ Processor MP Identification Information (Continued)

S-Spec	Core Stepping	Processor Signature	Speed Core/Data Bus (GHz/MHz)	L2 Cache Size	L3 Cache Size	Hyper-Threading Technology	Processor Interposer Revision	Package And Revision[2]	S-Spec Notes
SL6Z2 SL6Z7	B1	0F25h	2.50/400	512-KB	1-MB	Yes	01	603-pin micro-PGA interposer with 42.5 mm FC-BGA package	1, 2, 4 1, 2, 3,4
SL6YL SL6Z8	B1	0F25h	2.80/400	512-KB	2-MB	Yes	01	603-pin micro-PGA interposer with 42.5 mm FC-BGA package	1, 2,4 1, 2, 3, 4
SL79V	C0	0F26h	3/400	512-KB	4-MB	Yes	01	603-pin micro-PGA interposer with 42.5 mm FC-BGA package	1,2,4,5
SL79Z	C0	0F26h	2.70/400	512-KB	2-MB	Yes	01	603-pin micro-PGA interposer with 42.5 mm FC-BGA package	1, 2,4
SL7A5	C0	0F26h	2.20/400	512-KB	2-MB	Yes	01	603-pin micro-PGA interposer with 42.5 mm FC-BGA package	1, 2, 4

NOTES:
1. These parts require the inputs from A20M#, IGNNE#, LINT[1]/NMI and LINT[0]/INTR pins during RESET to set the correct core to bus frequency ratio.
2. The **Intel® Xeon™ Processor MP listed here is installed onto a micro pin grid array (mPGA) interposer. The overall processor package is called INT-mPGA.**
3. This part is an Intel boxed processor.
4. This part is the **Intel® Xeon™** Processor MP with up to 4-MB L3 cache on 0.13- micron process.
5. This part has a VID of 1.5V.

Mixed Steppings In MP Systems

Intel Corporation fully supports mixed steppings of Intel Xeon Processors MP. The following list and processor matrix describes the requirements to support mixed steppings:

- Mixed steppings are only supported with processors that have identical family numbers as indicated by the `CPUID` instruction. The Intel Xeon Processor MP is available with two different Model numbers as indicated by the `CPUID`. Please refer to the "Table 2 for details regarding inclusion of processors with mixed `CPUID`/Core steppings.
- While Intel has done nothing to specifically prevent processors operating at differing frequencies from functioning within a multiprocessor system, there may be uncharacterized errata that exist in such configurations. Intel does not support such configurations. In mixed stepping systems, all processors must operate at identical frequencies (i.e., the highest frequency rating commonly supported by all processors).
- While there are no known issues associated with the mixing of processors with differing cache sizes in a multiprocessor system, and Intel has done nothing to specifically prevent such system configurations from operating, Intel does not support such configurations since there may be uncharacterized errata that exist. In mixed stepping systems, all processors must be of the same cache size.
- While Intel believes that certain customers may wish to perform validation of system configurations with mixed frequency or cache sizes, and that those efforts are an acceptable option to our customers, customers would be fully responsible for the validation of such configurations.
- Intel requires that the proper microcode update be loaded on each processor operating in a multiprocessor system. Any processor that does not have the proper microcode update loaded is considered by Intel to be operating out of specification.
- The workarounds identified in this and following specification updates must be properly applied to each processor in the system. Certain errata are specific to the multiprocessor environment and are identified in Table 2 found at the end of this section. Errata for all processor steppings will affect system performance if not properly worked around. Also see Table 1 section for additional details on which processors are affected by specific errata.
- In mixed stepping systems, the processor with the lowest feature-set, as determined by the CPUID Feature Bytes, must be the bootstrap processor (BSP). In the event of a tie in feature-set, the tie should be resolved by selecting the BSP as the processor with the lowest stepping as determined by the `CPUID` instruction.

In the following processor matrix, **"NI"** indicates that there are currently no known issues associated with mixing these steppings. A number indicates that a known issue has been identified as listed in the table following the matrix. **"X"** indicates the processors cannot be mixed. A multiple processor system using mixed processor steppings must assure that errata are addressed appropriately for each processor.

Table 2. MP Platform Population Matrix for the Intel® Xeon™ Processor MP

Processor Signature/Core Stepping	0F11h/C0	0F22h/A0	0F25h/B1	0F26h/C0[2]	0F26h/C0[3]
0F11h/C0	NI	X	X	X	X
0F22h/A0	X	NI	NI	NI	X
0F25h/B1	X	NI	NI	NI	X
0F26h/C0[2]	X	NI	NI	NI	X
0F26h/C0[3]	X	X	X	X	NI

NOTES:
1. Some of these processors are affected by errata that may affect the features an MP system is able to support. See the Intel® Xeon™ Processor MP Errata table and Table 1 Information table for details on which processors are affected by these errata.
2. Only Applies to Stepping 0F26h with VID of 1.475 V.
3. Only Applies to Stepping 0F26h with VID of 1.5 V.

Summary Tables of Changes

The following table indicates the Errata, Documentation Changes, Specification Clarifications, or Specification Changes that apply to the Intel Xeon Processor MP. Intel intends to fix some of the errata in a future stepping of the component, and to account for the other outstanding issues through documentation or specification changes as noted. This table uses the following notation:

Codes Used In Summary Table

X:	Erratum, Specification Change or Clarification that applies to the given processor stepping.
(No mark) or (Blank Box):	This erratum is fixed in listed stepping or specification change does not apply to listed stepping.
Doc:	Document change or update that will be implemented.
Plan Fix:	This erratum may be fixed in a future of the product.
Fixed:	This erratum has been previously fixed.
No Fix:	There are no plans to fix this erratum.
PKG:	This column refers to errata on the Intel Xeon processor substrate.
AP	APIC-related erratum.
	Change bar to left of table row indicates this item is either new or modified from the previous version of this document.

Each Specification Update item will be prefixed with a capital letter to distinguish the product. The key below details the letters that are used in Intel's microprocessor Specification Updates:

A = Intel® Pentium® II processor
B = Mobile Intel® Pentium® II processor
C = Intel® Celeron® processor
D = Intel® Pentium® II Xeon™ processor
E = Intel® Pentium® III processor
F = Intel® Pentium® 4 processor Extreme Edition
G = Intel® Pentium® III Xeon™ processor
H = Mobile Intel® Celeron® processor at 466/433/400/366/333/300 and 266 MHz
K = Mobile Intel® Pentium® III processor
L = Intel® Celeron® D processor
M = Mobile Intel® Celeron® processor
N = Intel® Pentium® 4 processor
O = Intel® Xeon™ processor MP
P = Intel® Xeon™ processor
Q = Mobile Intel® Pentium® 4 processor supporting Hyper-Threading Technology on 90-nm process technology
R = Intel® Pentium® 4 processor on 90 nm process
S = 64-bit Intel® Xeon™ processor with 800 MHz system bus
T = Mobile Intel® Pentium® 4 processor-M
V = Mobile Intel® Celeron® processor on .13 Micron Process in Micro-FCPGA Package

W = Intel® Celeron® M processor
X = Intel® Pentium® M processor on 90 nm process with 2-MB L2 Cache
Y = Intel® Pentium® M processor
Z = Mobile Intel® Pentium® 4 processor with 533 MHz system bus

Note: The Specification Updates for the Pentium® processor, Pentium® Pro processor, and other Intel products do not use this convention.

Errata (Sheet 1 of 4)

No.	0F11h /C0	0F22h/ A0	OF25h\/ B0	0F26h/ C0	Plans	Errata
O1	X				Fixed	UC code in same line as write back (WB) data may lead to data corruption
O2	X	X	X	X	No Fix	Transaction is not retried after BINIT#
O3	X	X	X	X	No Fix	Invalid opcode 0FFFh requires a ModRM byte
O4	X	X	X	X	No Fix	FSW may not be completely restored after page-fault on FRSTOR or FLDENV instructions
O5	X	X	X	X	No Fix	Shutdown and IERR# may result due to a machine check exception on a Hyper-Threading Technology enabled processor
O6	X	X	X	X	No Fix	When in no-fill mode (CR0.CD=1) the memory type of large (PSE-4M and PAE-2M) pages are wrongly forced to uncacheable
O7	X	X	X	X	No Fix	Processor may hang due to speculative page walks to non-existent system memory
O8	X				Fixed	Writing a performance counter may result in an incorrect counter value
O9	X				Fixed	Performance counter may contain incorrect value after being stopped
O10	X	X	X	X	No Fix	Memory type of the load lock different from its corresponding store unlock
O11	X	X	X	X	No Fix	Machine check architecture error reporting and recovery may not work as expected
O12	X	X	X	X	No Fix	Debug mechanisms may not function as expected
O13	X				Fixed	Processor may timeout waiting for a device to respond after 0.67 seconds
O14	X	X	X	X	No Fix	Cascading of performance counters does not work correctly when forced overflow is enabled
O15	X	X	X	X	No Fix	EMON event counting of x87 loads may not work as expected
O16	X				Fixed	Simultaneous code breakpoint and uncorrectable error results in a processor hang
O17	X				Fixed	Software controlled clock modulation using a 12.5% or 25% duty cycle may cause the processor to hang
O18	X				Fixed	Processor samples bus frequency power-on configuration pins at the assertion of PWRGOOD
O19	X				Fixed	PAT index MSB may be calculated incorrectly
O20	X	X	X	X	No Fix	System bus interrupt messages without data which receive a hardfailure response may hang the processor
O21	X	X	X	X	No Fix	Bus invalidate line requests that return unexpected data may result in L1 cache corruption
O22	X	X	X	X	No Fix	The processor signals page-fault exception (#PF) instead of alignment check exception (#AC) on an unlocked CMPXCHG8B instruction
O23	X	X	X	X	No Fix	Incorrect data may be returned when page tables are located in write combining (WC) memory
O24	X				Fixed	Multiprocessor boot protocol may not complete with an IOQ depth of one
O25	X	X	X	X	No Fix	Write combining (WC) load may result in an unintended address on system bus

Errata (Sheet 2 of 4)

No.	0F11h /C0	0F22h/ A0	OF25h\/ B0	0F26h/ C0	Plans	Errata
O26	X	X	X	X	No Fix	Processor issues inconsistent transaction size attributes for locked operations
O27	X	X	X	X	No Fix	Multiple accesses to the same S-state L2 cache line and ECC error combination may result in loss of cache coherence
O28	X	X	X	X	No Fix	IA32_MC0_ADDR and IA32_MC0_MISC registers will contain invalid or stale data following a data, address, or response parity error
O29	X	X	X	X	No Fix	Instruction pointer stored on stack may become invalid
O30	X	X	X	X	No Fix	When the processor is in the system management mode (SMM), debug registers may be fully writeable
O31	X				Fixed	Associated counting logic must be configured when using event selection control (ESCR) MSR
O32	X	X			Fixed	Livelock may occur when bus parking is disabled
O33	X	X	X	X	No Fix	CPUID function 2 may return incorrect cache size information
O34	X	X	X	X	No Fix	CR2 may be incorrect or an incorrect page-fault error code may be pushed onto stack after execution of an LSS instruction
O35	X	X	X	X	No Fix	Hyper-Threading Technology enabled processors may hang in the presence of extensive self-modifying code
O36	X	X	X	X	No Fix	Global bit incorrectly set for secondary logical processors in ITLB
O37	X	X			Fixed	Hardware prefetcher may cause stale data to be loaded into the processor caches
O38	X	X	X	X	No Fix	System may hang if a fatal cache error causes bus write line (BWL) transaction to occur to the same cache line address as an outstanding bus read line (BRL) or bus read-invalidate line (BRIL)
O39	X	X	X	X	No Fix	Re-mapping the APIC base address to a value less than or equal to 0xDC001000 may cause I/O and special cycle failure
O40	X				Fixed	Erroneous machine check error reported
O41	X	X	X	X	No Fix	Processor does not flag #GP on non-zero write to certain MSRs
O42	X	X	X	X	No Fix	Counting both L2 and L3 cache reference events may result in undercount
O43	X	X	X	X	No Fix	Simultaneous assertion of A20M# and INIT# may result in incorrect data fetch
O44	X				Fixed	Incorrect Brand ID and Brand string
O45	X				Fixed	CPUID instruction returns incorrect number of ITLB entries
O46	X	X	X	X	No Fix	A write to APIC task priority register (TPR) that lowers priority may seem to have not occurred
O47		X			Fixed	Processor does not respond to break requests from ITP
O48		X	X	X	No Fix	Erroneous BIST result found in EAX register after reset
O49		X			Fixed	False data strobe glitch machine check error may occur when the machine check handler is enabled
O50		X	X	X	No Fix	Processor may hang under certain frequencies and 12.5% STPCLK# duty cycle
O51		X	X		Fixed	BPM[5:3]# VIL does not meet specification
O52	X	X	X	X	Plan Fix	STPCLK# signal assertion under certain conditions may cause a system hang

Errata (Sheet 3 of 4)

No.	0F11h /C0	0F22h/ A0	OF25h\/ B0	0F26h/ C0	Plans	Errata
O53	X	X	X	X	No Fix	Parity error in the L1 cache may cause the processor to hang
O54	X	X	X		Fixed	The TCK input in the test access port (TAP) is sensitive to low clock edge rates and prone to noise coupling onto TCK's rising or falling edges
O55	X	X	X	X	No Fix	Disabling a local APIC disables both logical processor APICs on a Hyper-Threading Technology enabled processor
O56	X	X	X	X	No Fix	Using STPCLK and executing code from very slow memory could lead to a system hang
O57			X	X	Plan Fix	Simultaneous cache line eviction from L2 and L3 caches may result in the write back of stale data
O58	X	X	X	X	No Fix	The state of the resume flag (RF flag) in a task-state segment (TSS) may be incorrect
O59	x				No Fix	Changes to CR3 register do not fence pending instruction page
O60	X	X	X	X	Plan Fix	Simultaneous page-faults at similar page offsets on both logical processors of an Hyper-Threading Technology enabled processor may cause application failure
O61	X	X	X	X	No Fix	A 16-bit address wrap resulting from a near branch (jump or call) may cause an incorrect address to be reported to the #GP exception handler
O62			X		Fixed	Incorrect PIROM L3 cache present value
O63	X	X	X	X	No Fix	Locks and SMC detection may cause the processor to temporarily hang
O64	X	X	X	X	No Fix	Incorrect debug exception (#DB) may occur when a data breakpoint is set on an FP instruction
O65			X	X	No Fix	Modified cache line eviction from L2 cache may result in write back of stale data
O66	X	X	X	X	No Fix	xAPIC may not report some illegal vector errors
O67	X	X	X	X	Plan Fix	Incorrect duty cycle is chosen when On-Demand Clock Modulation is enabled in a processor supporting Hyper-Threading Technology
O68	X	X	X	X	Plan Fix	Memory aliasing of pages as uncacheable memory type and write back (WB) may hang the system
O69			X	X	Plan Fix	A timing marginality in the Instruction Decoder unit may cause an unpredictable application behavior and/or system hang
O70	X	X	X	X	No Fix	Missing Stop Grant Acknowledge special bus cycle may cause a system hang
O71	X	X	X	X	No Fix	Machine check exceptions may not update Last-Exception Record MSRs (LERs)
O72	X	X	X	X	No Fix	Stores to page tables may not be visible to page walks for subsequent loads without serializing or invalidating the page table entry
O73			X	X	Plan Fix	A timing marginality in the Arithmetic Logic Unit (ALU) may cause indeterminate behavior
O74	X	X	X	X	No Fix	With Trap Flag (TF) asserted, FP instruction that triggers an unmasked FP exception may take single step trap before retirement of instruction
O75	X	X	X	X	No Fix	PDE/PTE loads and continuous locked updates to the same cache line may cause a system livelock

Errata (Sheet 4 of 4)

No.	0F11h /C0	0F22h/ A0	OF25h\/ B0	0F26h/ C0	Plans	Errata
O76	X	X	X	X	No Fix	Branch Trace Store (BTS) and Precise Event Based Sampling (PEBS) may update memory outside the BTS/PREBS buffer
O77	X	X	X	X	No Fix	Memory Ordering Failure may occur with Snoop Filtering Third-Party Agents after issuing and completing a BWIL (Bus Write Invalidate Line) or BLW (Bus Locked Write) transaction
O78	X	X	X	X	No Fix	Control Register 2 (CR2) can be updated during a REP MOVS/STOS instruction with Fast Strings enabled

Specification Changes

No.	SPECIFICATION CHANGES
	None for this revision of the Specification Update

Specification Clarifications

No.	SPECIFICATION CLARIFICATIONS
O1	Specification Clarification with respect to Time-Stamp Counter

Documentation Changes

No.	DOCUMENTATION CHANGES
	None for this revision of the Specification Update

Errata

O1 UC code in same line as write back (WB) data may lead to data corruption

Problem: This erratum occurs when both code (being accessed as uncacheable [UC] or write combining [WC]) and data (being accessed as write back [WB]) are placed in the same cache line. The UC fetch will cause the processor to self-snoop and generate an implicit WB. The data supplied by this implicit WB may be corrupted due to the way the processor is currently handling self-modifying code.

Implication: UC code located in the same cache line as WB data may lead to data corruption.

Workaround: UC or WC code should not be located in the same 64-byte cache line as any location that is being stored to with WB data.

Status: For the steppings effected, see the *Summary Table of Changes*.

O2 Transaction is not retried after BINIT#

Problem: If the first transaction of a locked sequence receives a HITM# and DEFER# during the snoop phase it should be retried and the locked sequence restarted. However, if BINIT# is also asserted during this transaction, the transaction will not be retried.

Implication: When this erratum occurs, locked transactions will not be retried.

Workaround: None at this time.

Status: For the steppings effected, see the *Summary Table of Changes*.

O3 Invalid opcode 0FFFh requires a ModRM byte

Problem: Some invalid opcodes require a ModRM byte and other following bytes, while others do not. The invalid opcode 0FFFh did not require a ModRM in previous generation microprocessors such as Pentium II or Pentium III processors, but it is required in the Intel Xeon Processor MP.

Implication: The use of an invalid opcode 0FFFh without the ModRM byte may result in a page or limit fault on the Intel Xeon Processor MP.

Workaround: To avoid this erratum use ModRM byte with invalid 0FFFh opcode.

Status: For the steppings effected, see the *Summary Table of Changes*.

O4 FSW may not be completely restored after page-fault on FRSTOR or FLDENV instructions

Problem: If the FPU operating environment or FPU state (operating environment and register stack) being loaded by an `FLDENV` or `FRSTOR` instruction wraps around a 64Kbyte or 4Gbyte boundary and a page-fault (#PF) or segment limit fault (#GP or #SS) occurs on the instruction near the wrap boundary, the upper byte of the FPU status word (FSW) might not be restored. If the fault handler does not restart program execution at the faulting instruction, stale data may exist in the FSW.

Implication: When this erratum occurs, stale data will exist in the FSW.

Workaround: Ensure that the FPU operating environment and FPU state do not cross 64Kbyte or 4Gbyte boundaries. Alternately, ensure that the page-fault handler restarts program execution at the faulting instruction after correcting the paging problem.

Status: For the steppings effected, see the *Summary Table of Changes*.

O5 Shutdown and IERR# may result due to a machine check exception on a Hyper-Threading Technology enabled processor

Problem: When a Machine Check Exception (MCE) occurs due to an internal error, both logical processors on a Hyper-Threading (HT) Technology enabled processor normally vector to the MCE handler. However, if one of the logical processors is in the “Wait for SIPI” state, that logical processor will not have a MCE handler and will shut down and assert IERR#.

Implication: A processor with a logical processor in the “Wait for SIPI” state will shut down when an MCE occurs on the other thread.

Workaround: None at this time.

Status: For the steppings effected, see the *Summary Table of Changes*.

O6 When in no-fill mode (CR0.CD=1) the memory type of large (PSE-4M and PAE-2M) pages are wrongly forced to uncacheable

Problem: When the processor is operating in No-Fill Mode (CR0.CD=1), the page miss hardware incorrectly forces the memory type of large (PSE-4M and PAE-2M) pages to UC memory type regardless of the MTRR settings. By forcing the memory type of these pages to UC, load operations, which should hit valid data in the L1 cache, are forced to load the data from system memory. Some applications will lose the performance advantage associated with the caching permitted by other memory types.

Implication: This erratum may result in some performance degradation when using no-fill mode with large pages.

Workaround: None at this time.

Status: For the steppings effected, see the *Summary Table of Changes*.

O7 Processor may hang due to speculative page walks to non-existent system memory

Problem: A load operation issued speculatively by the processor that misses the data translation lookaside buffer (DTLB) results in a page-walk. A branch instruction older than the load retires so that this load operation is now in the mispredicted branch path. Due to an internal boundary condition, in some instances the load is not cancelled before the page walk is issued.

The page miss handler (PMH) starts a speculative page-walk for the Load and issues a cacheable load of the page directory entry (PDE). This PDE loads returns data that points to a page table entry in UC memory. The PMH issues the PTE Load to UC space, which is issued on the system bus. No response comes back for this load PTE operation since the address is pointing to system memory that does not exist.

This load to non-existent system memory causes the processor to hang because other bus requests are queued up behind this UC PTE load which never gets a response. If the load was accessing valid system memory, the speculative page-walk would successfully complete and the processor would continue to make forward progress.

The boundary conditions to generate this erratum are more likely to occur with HT Technology enabled but may also occur with HT Technology disabled.

Implication: Processor may hang due to speculative page walks to non-existent system memory.

Workaround: Page directories and page tables in UC memory space must point to system memory that exists.

Status: For the steppings effected, see the *Summary Table of Changes*.

O8 Writing a performance counter may result in an incorrect counter value

Implication: Accessing a performance counter also enables the counter input so that writing one half of the counter can cause the other half to increment. When a performance counter is written and the event counter for the event being monitored is non-zero, the performance counter will be incremented by the value on that event counter. Because the upper eight bits of the performance counter are not written at the same time as the lower 32 bits, the increment due to the non-zero event counter may cause a carry to the upper bits such that the performance counter contains a value higher than what was written. The worst-case error caused by this can be about 4 billion counts.

Implication: When this erratum occurs, the performance counter will contain a different value from that which was written.

Workaround: If the performance counter is set to select a null event and the counter configuration control register (CCCR) for that counter has its compare bit set to zero, before the performance counter is written, this erratum will not occur.

Status: For the steppings effected, see the *Summary Table of Changes*.

O9 Performance counter may contain incorrect value after being stopped

Problem: If a performance counter is stopped on the precise internal clock cycle where the intermediate carry from the lower 32 bits of the counter to the upper eight bits occurs, the intermediate carry is lost.

Implication: When this erratum occurs, the performance counter may contain a value about 4 billion ($_2$ 32) less than it should.

Workaround: Since this erratum does not occur if the performance counters are read when running, a possible workaround is to read the counter before stopping it.

Status: For the steppings effected, see the *Summary Table of Changes*.

O10 Memory type of the load lock different from its corresponding store unlock

Problem: A use-once protocol is employed to ensure that the processor in a multi-agent system may access data that is loaded into its cache on a RFO operation at least once before it is snooped out by another agent. This protocol is necessary to avoid a multi-agent livelock scenario in which the processor cannot gain ownership of a line and modify it before that data is snooped out by another agent. In the case of this erratum, split load lock instructions incorrectly trigger the use-once protocol. A load lock operation accesses data that splits across a page boundary with both pages of WB memory type. The use-once protocol activates and the memory type for the split halves get forced to UC. Since use-once does not apply to stores, the store unlock instructions go out as WB memory type. The full sequence on the bus is: locked partial read (UC), partial read (UC), partial write (WB), locked partial write (WB). The use-once protocol should not be applied to load locks.

Implication: When this erratum occurs, the memory type of the load lock will be different than the memory type of the store unlock operation. This behavior (load locks and store unlocks having different memory types) does not introduce any functional failures such as system hangs or memory corruption.

Workaround: None at this time

Status: For the steppings affected, see the Summary Tables of Changes.

O11 Machine check architecture error reporting and recovery may not work as expected

Problem: When the processor detects errors it should attempt to report and/or recover from the error. In the situations described below, the processor does not report and/or recover from the error(s) as intended.

- When a transaction is deferred during the snoop phase and subsequently receives a hard failure response, the transaction should be removed from the bus queue so that the processor may

proceed. Instead, the transaction is not properly removed from the bus queue, the bus queue is blocked, and the processor will hang.

- When a hardware prefetch results in an uncorrectable tag error in the L2 cache, MC0_STATUS.UNCOR and MC0_STATUS.PCC are set but no machine check exception (MCE) is signaled. No data loss or corruption occurs because the data being prefetched has not been used. If the data location with the uncorrectable tag error is subsequently accessed, an MCE will occur. However, upon this MCE, or any other subsequent MCE,.the information for that error will not be logged because MC0_STATUS.UNCOR has already been set and the MCA status registers will not contain information about the error which caused the MCE assertion but instead will contain information about the prefetch error event.
- When the reporting of errors is disabled for machine check architecture (MCA) Bank 2 by setting all MC2_CTL register bits to 0, uncorrectable errors should be logged in the IA32_MC2_STATUS register but no machine-check exception should be generated. Uncorrectable loads on bank 2, which would normally be logged in the IA32_MC2_STATUS register, are not logged.
- When one half of a 64 byte instruction fetch from the L2 cache has an uncorrectable error and the other 32 byte half of the same fetch from the L2 cache has a correctable error, the processor will attempt to correct the correctable error but cannot proceed due to the uncorrectable error. When this occurs the processor will hang.
- When an L1 cache parity error occurs, the cache controller logic should write the physical address of the data memory location that produced that error into the IA32_MC1_ADDR REGISTER (MC1_ADDR). In some instances of a parity error on a load operation that hits the L1 cache, the cache controller logic may write the physical address from a subsequent load or store operation into the IA32_MC1_ADDR register.
- When an error exists in the tag field of a cache line such that a Request For Ownership (RFO) issued by the processor hits multiple tag fields in the L2 cache (the correct tag and the tag with the error) and the accessed data also has a correctable error, the processor will correctly log the multiple tag match error but will hang when attempting to execute the machine check exception handler.
- If a memory access receives a machine check error on both 64 byte halves of a 128-byte L2 cache sector, the IA32_MC0_STATUS register records this event as multiple errors, i.e., the valid error bit and the overflow error bit are both set indicating that a machine check error occurred while the results of a previous error were in the error-reporting bank. The IA32_MC1_STATUS register should also record this event as multiple errors but instead records this event as only one correctable error.
- The overflow bit should be set to indicate when more than one error has occurred. The overflow bit being set indicates that more than one error has occurred. Because of this erratum, if any further errors occur, the MCA overflow bit will not be updated, thereby incorrectly indicating only one error has been received.
- If an I/O instruction (IN, INS, REP INS, OUT, OUTS, or REP OUTS) is being executed, and if the data for this instruction becomes corrupted, the processor will signal a MCE. If the instruction is directed at a device that is powered down, the processor may also receive an assertion of SMI#. Since MCEs have higher priority, the processor will call the MCE handler, and the SMI# assertion will remain pending. However, while attempting to execute the first instruction of the MCE handler, the SMI# will be recognized and the processor will attempt to execute the SMM handler. If the SMM handler is successfully completed, it will attempt to restart the I/O instruction, but will not have the correct machine state due to the call to the MCE handler. This can lead to failure of the restart and shutdown of the processor.
- If PWRGOOD is de-asserted during a RESET# assertion causing internal glitches, the MCA registers may latch invalid information.

- If RESET# is asserted, then de-asserted, and reasserted, before the processor has cleared the MCA registers, then the information in the MCA registers may not be reliable, regardless of the state or state transitions of PWRGOOD.
- If MCERR# is asserted by one processor and observed by another processor, the observing processor does not log the assertion of MCERR#. The MCE handler called upon assertion of MCERR# will not have any way to determine the cause of the MCE.
- The Overflow Error bit (bit 62) in the IA32_MC0_STATUS register indicates, when set, that a machine check error occurred while the results of a previous error were still in the error reporting bank (i.e., The Valid bit was set when the new error occurred). If an uncorrectable error is logged in the error-reporting bank and another error occurs, the overflow bit will not be set.
- The MCA Error Code field of the IA32_MC0_STATUS register gets written by a different mechanism than the rest of the register. For uncorrectable errors, the other fields in the IA32_MC0_STATUS register are only updated by the first error. Any further errors that are detected will update the MCA Error Code field without updating the rest of the register, thereby leaving the IA32_MC0_STATUS register with stale information.
- When a speculative load operation hits the L2 cache and receives a correctable error, the IA32_MC1_Status Register may be updated with incorrect information. The IA32_MC1_Status Register should not be updated for speculative loads.
- The processor should only log the address for L1 parity errors in the IA32_MC1_Status register if a valid address is available. If a valid address is not available, the Address Valid bit in the IA32_MC1_Status register should not be set. In instances where an L1 parity error occurs and the address is not available because the linear to physical address translation is not complete or an internal resource conflict has occurred, the Address Valid bit is incorrectly set.
- The processor may hang when an instruction code fetch receives a hard failure response from the system bus. This occurs because the bus control logic does not return data to the core, leaving the processor empty. IA32_MC0_STATUS MSR does indicate that a hard fail response occurred.

The processor may hang when the following events occur and the machine check exception is enabled, CR4.MCE=1. A processor that has it's STPCLK# pin asserted will internally enter the Stop Grant State and finally issue a Stop Grant Acknowledge special cycle to the bus. If an uncorrectable error is generated during the Stop Grant process it is possible for the Stop Grant special cycle to be issued to the bus before the processor vectors to the machine check handler. Once the chipset receives its last Stop Grant special cycle it is allowed to ignore any bus activity from the processors. As a result, processor accesses to the machine check handler may not be acknowledged, resulting in a processor hang.

Implication: The processor is unable to correctly report and/or recover from certain errors.

Workaround: None at this time.

Status: For the steppings effected, see the *Summary Table of Changes*.

O12 Debug mechanisms may not function as expected

Problem: Certain debug mechanisms may not function as expected on the processor. The cases are as follows:

- When the following conditions occur: 1) An FLD instruction signals a stack overflow or underflow, 2) the FLD instruction splits a page-boundary or a 64 byte cache line boundary, 3) the instruction matches a debug register on the high page or cache line respectively, and 4) the FLD has a stack fault and a memory fault on a split access, the processor will only signal the stack fault and the debug exception will not be taken.

- When a data breakpoint is set on the ninth and/or tenth byte(s) of a floating point store using the Extended Real data type, and an unmasked floating point exception occurs on the store, the break point will not be captured.
- When any instruction has multiple debug register matches, and any one of those debug registers is enabled in DR7, all of the matches should be reported in DR6 when the processor goes to the debug handler. This is not true during a `REP` instruction. As an example, during execution of a `REP MOVSW` instruction the first iteration a load matches DR0 and DR2 and sets DR6 as FFFF0FF5h. On a subsequent iteration of the instruction, a load matches only DR0. The DR6 register is expected to still contain FFFF0FF5h, but the processor will update DR6 to FFFF0FF1h.

A Data breakpoint that is set on a load to UC memory may be ignored due to an internal segment register access conflict. In this case the system will continue to execute instructions, bypassing the intended breakpoint. Avoiding having instructions that access segment descriptor registers, e.g., LGDT, LIDT close to the UC load, and avoiding serialized instructions before the UC load will reduce the occurrence of this erratum.

Implication: Certain debug mechanisms do not function as expected on the processor.

Workaround: None at this time.

Status: For the steppings effected, see the *Summary Table of Changes*.

O13 Processor may timeout waiting for a device to respond after 0.67 seconds

Problem: The PCI 2.1 target initial latency specification allows two seconds for a device to respond during initialization-time. The processor may timeout after only approximately 0.67 seconds. When the processor times out it will hang with IERR# asserted. PCI devices that take longer than 0.67 seconds to initialize may not be initialized properly.

Implication: System may hang with IERR# asserted.

Workaround: Due to the long initialization time observed on some commercially available PCI cards, it may be necessary to disable the timeout counter during the PCI initialization sequence. This can be accomplished by temporarily setting Bit 5 of the MISC_ENABLES_MSR located at address 1A0H to 1 for all processor in the system. This model specific register (MSR) is software visible but should only be set for the duration of the PCI initialization sequence. It is necessary to re-enable the timeout counter by clearing this bit after completing the PCI initialization sequence. CAUTION: The processor's Thermal Monitor feature may not function if the timeout counter is not re-enabled after completing the PCI initialization.

After the system is fully initialized, this erratum may occur either when a PCI device is hot added into the system or when a PCI device is transitioned from D3 cold. System software responsible for completing the hot add and the power state transition from D3 cold should allow for a delay of the target initial latency prior to initiating configuration accesses to the PCI device.

Status: For the steppings effected, see the *Summary Table of Changes*.

O14 Cascading of performance counters does not work correctly when forced overflow is enabled

Problem: The performance counters are organized into pairs. When the CASCADE bit of the CCCR is set, a counter that overflows will continue to count in the other counter of the pair. The FORCE_OVF bit forces the counters to overflow on every non-zero increment. When the FORCE_OVF bit is set, the counter overflow bit will be set but the counter no longer cascades.

Implication: The performance counters do not cascade when the FORCE_OVF bit is set.

Workaround: None at this time.

Status: For the steppings effected, see the *Summary Table of Changes*.

O15 EMON event counting of x87 loads may not work as expected

Problem: If a performance counter is set to count x87 loads and floating-point exceptions are unmasked, the FPU Operand (Data) Pointer (FDP) may become corrupted.

Implication: When this erratum occurs, the FDP may become corrupted.

Workaround: This erratum will not occur with floating-point exceptions masked. If floating-point exceptions are unmasked, then performance counting of x87 loads should be disabled.

Status: For the steppings effected, see the *Summary Table of Changes*.

O16 Simultaneous code breakpoint and uncorrectable error results in a processor hang

Problem: If an instruction fetch results in an uncorrectable error and there is also a debug breakpoint at this address, the processor will hang and the uncorrectable error will not be logged in the Machine Check registers.

Implication: When this erratum occurs the processor will livelock.

Workaround: None at this time.

Status: For the steppings effected, see the *Summary Table of Changes*.

O17 Software controlled clock modulation using a 12.5% or 25% duty cycle may cause the processor to hang

Problem: Processor clock modulation may be controlled via a processor register (IA32_THERM_CONTROL). The On-Demand Clock Modulation Duty Cycle is controlled by bits 3:1. If these bits are set to a duty cycle of 12.5% or 25%, the processor may hang while attempting to execute a floating-point instruction. In this failure, the last instruction pointer (LIP) is pointing to a floating-point instruction whose instruction bytes are in UC space and which takes a floating-point error exception. The processor continuously cycles attempting to fetch the bytes of the faulting floating-point instruction and those following it. This erratum is caused by interactions between the thermal control circuit and floating-point event handler.

Implication: When software controlled clock modulation is used with a duty cycle of 12.5% or 25% the processor will go into a sleep state from which it fails to return.

Workaround: Use a duty cycle other than 12.5% or 25%.

Status: For the steppings effected, see the *Summary Table of Changes*.

O18 Processor samples bus frequency power-on configuration pins at the assertion of PWRGOOD

Problem: According to the *Intel® Xeon™ Processor MP Electrical, Mechanical, and Thermal Specifications (EMTS)*, the bus frequency-to-core ratio may be set by the power-on configuration option pins LINT[1:0], IGNNE#, and A20M#. The processor should only sample these pins on the active-to-inactive transition of RESET#, however, the processor is also sampling these pins on the inactive-to-active transition of PWRGOOD. The internal initialization done by the processor between the assertion of PWRGOOD and the deassertion of RESET# may be affected if this ratio represents a high frequency at which the part will not properly function. This failure to initialize the processor properly may prevent the processor from coming out of reset or prevent some features such as the thermal control circuit from working properly.

Implication: The processor may fail to initialize properly if the frequency specified by the power-on configuration bits sampled at the assertion of PWRGOOD is too high for the processor to function correctly. On production parts and qualification samples, the frequency is internally limited so that this erratum should have no impact.

Workaround: No workaround is required for systems using qualification or production processors.

Status: For the steppings effected, see the *Summary Table of Changes*.

O19 PAT index MSB may be calculated incorrectly

Problem: When Mode C or Mode B paging support is enabled and all of the following events occur:

- A page walk returns the page directory entry (PDE) for a large page from memory.
- A subsequent page walk returns the page table entry (PTE) for a 4k page from memory and the page attribute table (PAT) upper index bit (bit 7) in this PTE is set to 1b.

It is possible that the PAT upper index bit in the PTE is incorrectly ignored and assumed to be 0b. The result is that the memory type in the PAT that should have come from the corresponding PAT index [4-7] incorrectly comes from PAT index [0-3].

Implication: If an operating system has programmed the PAT in an asymmetrical fashion i.e., PAT[0-3] is different from PAT[4-7] then an incorrect memory type may be used.

Workaround: None at this time.

Status: For the steppings effected, see the *Summary Table of Changes*.

O20 System bus interrupt messages without data which receive a hardfailure response may hang the processor

Problem: When a system bus agent (processor or chipset) issues an interrupt transaction without data onto the system bus and the transaction receives a HardFailure response, a potential processor hang can occur. The processor, which generates an inter-processor interrupt (IPI) that receives the HardFailure response, will still log the MCA error event cause as HardFailure, even if the APIC causes a hang. Other processors, which are true targets of the IPI, will also hang on hardfail-without-data, but will not record an MCA HardFailure event as the cause. If a HardFailure response occurs on a system bus interrupt message with data, the APIC will complete the operation so as not to hang the processor.

Implication: The processor may hang.

Workaround: None at this time.

Status: For the steppings affected, see the Summary Tables of Changes.

O21 Bus invalidate line requests that return unexpected data may result in L1 cache corruption

Problem: When a bus invalidate line (BIL) request receives unexpected data from a deferred reply, and a store operation write combines to the same address, there is a small window where the L1cache is corrupt, and loads can retire with this corrupted data. This erratum occurs in the following scenario:

- A RFO transaction is issued by the processor and hits a line in shared state in the L2 cache.
- The RFO is then issued on the system bus as a 0 length read-invalidate (BIL), since it doesn't need data, just ownership of the cache line.
- This transaction is deferred by the chipset.
- At some later point, the chipset sends a deferred reply for this transaction with an implicit write-back response. For this erratum to occur, no snoop of this cache line can be issued between the BIL and the deferred reply.
- The processor issues a write-combining store to the same cache line while data is returning to the processor. This store straddles an 8-byte boundary.
- Due to an internal boundary condition, a time window exists where the L1 cache contains corrupt data which could be accessed by a load.

Implication: The L1 cache may contain corrupted data. No known commercially available chipsets trigger the failure conditions.

Workaround: The chipset could issue a BIL (snoop) to the deferred processor to eliminate the failure conditions.

Status: For the steppings effected, see the *Summary Table of Changes*.

O22 The processor signals page-fault exception (#PF) instead of alignment check exception (#AC) on an unlocked CMPXCHG8B instruction

Problem: If a page-fault exception (#PF) and alignment check exception (#AC) both occur for an unlocked `CMPXCHG8B` instruction, then #PF will be flagged.

Implication: Software that depends on #AC before the #PF will be affected since #PF is signaled in this case.

Workaround: Remove the software's dependency on #AC having precedence over #PF. Alternately, correct the page-fault in the page-fault handler and then restart the faulting instruction.

Status: For the steppings effected, see the *Summary Table of Changes*.

O23 Incorrect data may be returned when page tables are located in write combining (WC) memory

Problem: If page directories and/or page tables are located in WC memory, speculative loads to cacheable memory may complete with incorrect data.

Implication: Cacheable loads to memory mapped using page tables located in WC memory may return incorrect data. Intel has not been able to reproduce this erratum with commercially available software.

Workaround: Do not place page directories and/or page tables in WC memory.

Status: For the steppings effected, see the *Summary Table of Changes*.

O24 Multiprocessor boot protocol may not complete with an IOQ depth of one

Problem: When the in-order queue (IOQ) depth is managed by the chipset to be one entry deep, the system may hang during the multi-processor boot protocol. This hang occurs when the chipset drives BNR# in such a way that the processors are continually throttled off the bus then released to access the bus in alternating cycles which never allows the multi-processor boot protocol to complete execution.

Implication: The system may hang during the multiprocessor boot protocol.

Workaround: If the chipset drives BNR# in such a way that the processors are continually throttled off the bus then released to access the bus in alternating cycles, do not use IOQ de-pipelining.

Status: For the steppings effected, see the *Summary Table of Changes*.

O25 Write combining (WC) load may result in an unintended address on system bus

Problem: When the processor performs a speculative WC load, down the path of a mispredicted branch, and the address happens to match a valid UC address translation with the DTLB, an unintended UC load operation may be sent out on the system bus.

Implication: When this erratum occurs, an unintended load may be sent on the system bus. Intel has only encountered this erratum during pre-silicon simulation.

Workaround: It is possible for the BIOS to contain a workaround for this erratum for some steppings of the processor.

Status: For the steppings effected, see the *Summary Table of Changes*.

O26 Processor issues inconsistent transaction size attributes for locked operations

Problem: When the processor is in the page address extension (PAE) mode and detects the need to set the Access and/or Dirty bits in the page directory or page table entries, the processor sends an 8 byte load lock onto the system bus. A subsequent 8 byte store unlock is expected, but instead a 4 byte store unlock occurs. Correct data is provided since only the lower bytes change, however external logic monitoring the data transfer may be expecting an 8-byte store unlock.

Implication: No known commercially available chipset are affected by this erratum.

Workaround: None at this time.

Status: For the steppings effected, see the *Summary Table of Changes*.

O27 Multiple accesses to the same S-state L2 cache line and ECC error combination may result in loss of cache coherence

Problem: When a RFO cycle has a 64 bit address match with an outstanding read hit on a line in the L2 cache which is in the S-state AND that line contains an ECC error, the processor should recycle the RFO until the ECC error is handled. Due to this erratum, the processor does not recycle the RFO and attempt to service both the RFO and the read hit at the same time.

Implication: When this erratum occurs, cache may become incoherent.

Workaround: None at this time.

Status: For the steppings effected, see the *Summary Table of Changes*.

O28 IA32_MC0_ADDR and IA32_MC0_MISC registers will contain invalid or stale data following a data, address, or response parity error

Problem: If the processor experiences a data, address, or response parity error, the ADDRV and MISCV bits of the IA32_MC0_STATUS register are set, but the IA32_MC0_ADDR and IA32_MC0_MISC registers are not loaded with data regarding the error.

Implication: When this erratum occurs, the IA32_MC0_ADDR and IA32_MC0_MISC registers will contain invalid or stale data.

Workaround: Ignore any information in the IA32_MC0_ADDR and IA32_MC0_MISC registers after a data, address or response parity error.

Status: For the steppings effected, see the *Summary Table of Changes*.

O29 Instruction pointer stored on stack may become invalid

Problem: Due to an internal boundary condition which may exist on a HT Technology enabled Intel Xeon processor MP, the following sequence of events must occur:

Caution: One logical processor executes the `WRMSR` instruction with incorrect data causing a general protection fault.

1. Simultaneously an event that requires micro-architectural synchronization among the two logical processors occurs on the second logical processor may cause an invalid instruction pointer to be stored on the ring 0 stack during the transition to GP fault handler on the first logical processor.

Implication: The instruction pointer stored on the stack may be invalid, potentially causing errors during execution of or return from the GP fault handler.

Workaround: It is possible for BIOS to contain a workaround this issue. For HT Technology enabled processors; insure all `WRMSR` instructions do not generate GP faults due to incorrect data.

Status: For the steppings effected, see the *Summary Table of Changes*.

O30 When the processor is in the system management mode (SMM), debug registers may be fully writeable

Problem: When in system management mode (SMM), the processor executes code and stores data in the SMRAM space. When the processor is in this mode and writes are made to DR6 and DR7, the processor should block writes to the reserved bit locations. Due to this erratum, the processor may not block these writes. This may result in invalid data in the reserved bit locations.

Implication: Reserved bit locations within DR6 and DR7 may become invalid.

Workaround: Software may perform a read/modify/write when writing to DR6 and DR7 to ensure that the values in the reserved bits are maintained.

Status: For the steppings effected, see the *Summary Table of Changes*.

O31 Associated counting logic must be configured when using event selection control (ESCR) MSR

Problem: ESCR MSRs allow software to select specific events to be counted, with each ESCR usually associated with a pair of performance counters. ESCRs may also be used to qualify the detection of at-retirement events that support precise-event-based sampling (PEBS). A number of performance metrics that support PEBS require a 2nd ESCR to tag uops for the qualification of at-retirement events. (The first ESCR is required to program the at-retirement event.) Counting is enabled via counter configuration control registers (CCCR) while the event count is read from one of the associated counters. When counting logic is configured for the subset of at-retirement events that require a 2nd ESCR to tag uops, at least one of the CCCRs in the same group of the 2nd ESCR must be enabled.

Implication: If no CCCR/counter is enabled in a given group, the ESCR in that group that is programmed for tagging uops will have no effect. Hence a subset of performance metrics that require a 2nd ESCR for tagging uops may result in 0 count.

Workaround: Ensure that at least one CCCR/counter in the same group as the tagging ESCR is enabled for those performance metrics that require two ESCRs and tagging uops for at-retirement counting.

Status: For the steppings effected, see the *Summary Table of Changes*.

O32 Livelock may occur when bus parking is disabled

Problem: A livelock may occur when processor bus parking is disabled, and when (1) the processor is the symmetric owner of the bus with one internal request pending, and (2) the processor observes the assertion of BPRI#, BNR# or a full IOQ. In this scenario, the processor bus interface unit assumes that the assertion of ADS# is not required, deasserts BREQ, and, as a result, relinquishes bus ownership without issuing the pending request. If the BPRI#, BNR# or full IOQ pattern continues coincident with the arbitration phase of the processor that still has only one outstanding internal request, livelock may occur. Assertion of bus parking, any change to the regular pattern of BPRI# or BNR# assertion noted above, or the arrival of a second internal transaction will release the processor from the livelock condition.

Implication: This erratum may result in a livelock.

Workaround: This erratum can be avoided by enabling bus parking. The deassertion of signal A15# during the active-to-inactive edge of RESET# will enable bus parking.

Status: For the steppings effected, see the *Summary Table of Changes*.

O33 CPUID function 2 may return incorrect cache size information

Problem: When a HT Technology-enabled processor executes a `CPUID` instruction with function 2 (02 in the EAX register), the processor may return incorrect/invalid cache descriptors in the EDX register. Code must be executing on both logical processors to encounter this erratum.

Implication: When this erratum occurs the data returned to the EDX register may be inaccurate/invalid.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings effected, see the *Summary Table of Changes*.

O34 CR2 may be incorrect or an incorrect page-fault error code may be pushed onto stack after execution of an LSS instruction

Problem: Under certain timing conditions, the internal load of the selector portion of the LSS instruction may complete with potentially incorrect speculative data before the load of the offset portion of the address completes. The incorrect data is corrected before the completion of the LSS instruction but the value of CR2 and the error code pushed on the stack are reflective of the speculative state. Intel has not observed this erratum with commercially available software.

Implication: When this erratum occurs, the contents of CR2 may be off by two, or an incorrect page-fault error code may be pushed onto the stack.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings effected, see the *Summary Table of Changes*.

O35 Hyper-Threading Technology enabled processors may hang in the presence of extensive self-modifying code

Problem: For multiprocessor platforms, in which HT Technology enabled processors are executing extensive self modifying code, and branch trace messages are enabled on at least one logical processor, the system may hang. In this scenario, a processor executing within 1K of code being written to by another processor may attempt to end this flow, thereby resulting in a hang.

Implication: When this erratum occurs the system will hang.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings effected, see the *Summary Table of Changes*.

O36 Global bit incorrectly set for secondary logical processors in ITLB

Problem: Due to a boundary condition in the translation look-aside buffer logic, the global bit information in the TLB entry for a mapping belonging to the first logical processor can overwrite the global bit information for a mapping belonging to the second logical processor. This occurs in the following scenario:

- The first logical processor misses the ITLB resulting in a page walk.
- The second logical processor also misses the ITLB and generates a page walk.

In certain timing scenarios within the processor, the leftover global bit information from the first logical processor may overwrite the second logical processor.

Implication: When this erratum occurs, if the page global bit for the second logical processor is overwritten with a 0b, this will result in performance degradation for the first logical processor. If the page global bit is incorrectly changed from a 0 to 1, this erratum may result in software failures.

Workaround: It is possible for BIOS code to contain a workaround for this erratum.

Status: For the steppings effected, see the *Summary Table of Changes*.

O37 Hardware prefetcher may cause stale data to be loaded into the processor caches

Problem: The processor may use stale data from the cache while the hardware prefetcher is enabled. The conditions of this erratum are as follows:

- A cache line is stored in the L3 cache in shared state while its adjacent sector is in modified state. The same cache line and its adjacent sector reside in the L2 cache in the shared and invalid state, respectively. The cache line and its adjacent sector are being evicted from the L3

cache at the same time that a prefetch RFO is issued to this address. A boundary condition exists in the bus logic where the prefetch may be issued on the system bus before the modified data in the L3 is written back to main memory. Consequently the RFO gets stale data for the adjacent sector from main memory and fills the cache with this stale data.

Implication: The processor may use stale data from the cache.

Workaround: Disable the Hardware Prefetcher by setting bit 9 in register IA32_MISC_ENABLE - MSR Address 01A0h via the BIOS.

Status: For the steppings effected, see the *Summary Table of Changes*.

O38 System may hang if a fatal cache error causes bus write line (BWL) transaction to occur to the same cache line address as an outstanding bus read line (BRL) or bus read-invalidate line (BRIL)

Problem: A processor internal cache fatal data ECC error may cause the processor to issue a BWL transaction to the same cache line address as an outstanding BRL or BRIL. As it is not typical behavior for a single processor to have a BWL and a BRL/BRIL concurrently outstanding to the same address, this may represent an unexpected scenario to system logic within the chipset.

Implication: The processor may not be able to fully execute the machine check handler in response to the fatal cache error if system logic does not ensure forward progress on the system bus under this scenario.

Workaround: System logic should ensure completion of the outstanding transactions. Note that during recovery from a fatal data ECC error, memory image coherency of the BWL with respect to BRL/BRIL transactions is not important. Forward progress is the primary requirement.

Status: For the steppings effected, see the *Summary Table of Changes*.

O39 Re-mapping the APIC base address to a value less than or equal to 0xDC001000 may cause I/O and special cycle failure

Problem: Re-mapping the APIC base address from its default can cause conflicts with either I/O or special cycle bus transactions.

Implication: Either I/O or special cycle bus transactions can be redirected to the APIC, instead of appearing on the front side bus.

Workaround: Use any APIC base addresses above 0xDC001000 as the relocation address.

Status: For the steppings effected, see the *Summary Table of Changes*.

O40 Erroneous machine check error reported

Problem: An erroneous multi-bit ECC machine check error may occur on HT Technology enabled processors when both logical processors are in the halt state. In this state, the clocks inside the processor will be shut off. There is a boundary case where a speculative page walk could be occurring while the clocks are shut off. This page walk continues after the clocks are turned back on. If the clocks are off when the page walk is in a specific pipe stage in the machine, an erroneous L2 tag ECC error may be observed.

Implication: Due to this erratum, an erroneous multi-bit error may be reported in the machine check registers when machine check is enabled. There is no known impact when machine check is disabled. There have been no observances of data corruption caused by this issue.

Workaround: It is possible for BIOS to contain a workaround for this issue.

Status: For the steppings effected, see the *Summary Table of Changes*.

O41 Processor does not flag #GP on non-zero write to certain MSRs

Problem: When a non-zero write occurs to the upper 32 bits of IA32_CR_SYSENTER_EIP or IA32_CR_SYSENTER_ESP, the processor should indicate a general protection fault by flagging #GP. Due to this erratum, the processor does not flag #GP.

Implication: The processor unexpectedly does not flag #GP on a non-zero write to the upper 32 bits of IA32_CR_SYSENTER_EIP or IA32_CR_SYSENTER_ESP. No known commercially available operating system has been identified to be affected by this erratum.

Workaround: None at this time.

Status: For the steppings effected, see the *Summary Table of Changes*.

O42 Counting both L2 and L3 cache reference events may result in undercount

Problem: The processor's counting logic combines events from misaligned pipestages. Thus, if two requests are sent referencing L2 and L3 at the same time, the EMON logic may only log them as one event.

Implication: This may result in undercounting of cache reference events.

Workaround: Restrict to counting either L2 or L3 events, but not both at the same time on a single ESCR.

Status: For the steppings effected, see the *Summary Table of Changes*.

O43 Simultaneous assertion of A20M# and INIT# may result in incorrect data fetch

Problem: If A20M# and INIT# are simultaneously asserted by software, followed by a data access to the 0xFFFFFXXX memory region, with A20M# still asserted, incorrect data will be accessed. With A20M# asserted, an access to 0xFFFFFXXX should result in a load from physical address 0xFFEFFXXX. However, in the case of A20M# and INIT# being asserted together, the data load will actually be from the physical address 0xFFFFFXXX. Code accesses are not affected by this erratum.

Implication: Processor may fetch incorrect data, resulting in BIOS failure.

Workaround: Deasserting and reasserting A20M# prior to the data access will workaround this erratum.

Status: For the steppings effected, see the *Summary Table of Changes*.

O44 Incorrect Brand ID and Brand string

Problem: The Brand ID for the production (S-Spec) Intel(R) Xeon(TM) processor MP processors should be 0Bh, which is associated with the Intel-branded text string "Intel(R) Xeon(TM) Processor MP". The Brand ID returned by all production (S-Spec) Intel(R) Xeon(TM) processor MP is 0Eh, which is associated with the Intel branded text string "Intel(R) Xeon(TM) Processor". In addition the Brand String extensions to the `CPUID` instruction also return the incorrect brand string, "Intel(R) Xeon(TM) CPU x.sixth".

Brand ID is a processor identification feature that is accessible via the `CPUID` instruction. Processors that implement the Brand ID feature return an 8-bit value in bits 7 through 0 of the EBX register when the `CPUID` instruction is executed with EAX=1. A full description of the Brand ID feature and a table of Brand ID values returned by various processors is included in the *Intel® Processor Identification and the CPUID Instruction (AP-485)* Application Note (see http://developer.intel.com/design/xeon/applnots/241618.htm).

Implication: Intel expects the impact of this issue to be limited to incorrect processor identification on BIOS POST or OS system information screens. However, BIOS developers and OEM system board manufacturers should judge the impact of this Brand ID issue on their existing platform designs incorporating the Intel Xeon processor MP.

Workaround: Refer to the Processor Signature portion of the `CPUID` instruction when determining processor brand. A processor signature of 0F11h = Intel(R) Xeon(TM) processor MP.

Status: For the steppings effected, see the *Summary Table of Changes*.

O45 CPUID instruction returns incorrect number of ITLB entries

Problem: When the `CPUID` instruction is executed with EAX = 2 on a processor without HT Technology or with HT Technology disabled via power on configuration, it should return a value of 51h in

EAX[15:8] to indicate that the instruction translation lookaside buffer (ITLB) has 128 entries. On a processor with HT Technology enabled, the processor should return 50h (64 entries). Due to this erratum, the CPUID instruction always returns 50h (64 entries).

Implication: Software may incorrectly report the number of ITLB entries. Operation of the processor is not affected.

Workaround: None at this time.

Status: For the steppings effected, see the *Summary Table of Changes*.

O46 A write to APIC task priority register (TPR) that lowers priority may seem to have not occurred

Problem: In respect to the retirement of instructions, stores to the UC memory-based APIC register space are handled in a non-synchronized way. For example if an instruction that masks the interrupt flag, e.g. CLI, is executed soon after an UC write to the task priority register (TPR) that lowers the APIC priority, the interrupt masking operation may take effect before the actual priority has been lowered. This may cause interrupts whose priority is lower than the initial TPR, but higher than the final TPR, to not be serviced until the interrupt flag is finally cleared, i.e. by STI instruction. Interrupts will remain pending and are not lost.

Implication: This condition may allow interrupts to be accepted by the processor but may delay their service**.**

Workaround: This non-synchronization can be avoided by issuing an APIC register read after the APIC register write. This will force the store to the APIC register before any subsequent instructions are executed. No commercial operating system is known to be impacted by this erratum.

Status: For the steppings effected, see the *Summary Table of Changes*.

O47 Processor does not respond to break requests from ITP

Problem: On power-up and low-power state transitions, the processor's TAP circuitry may remain in the tap-logic-reset (TLR) state.

Implication: The ITP is unable to cause a break on reset in the processor, which may prevent the loading of processor and chipset registers, or affect the ability to debug from cold boot and low power transitions.

Workaround: None at this time.

Status: For the steppings effected, see the *Summary Table of Changes*.

O48 Erroneous BIST result found in EAX register after reset

Problem: The processor may show an erroneous built-in self test (BIST) result in the EAX register bit 0 after reset.

Implication: When this erratum occurs, an erroneous BIST failure will be reported in the EAX register bit 0, however this failure can be ignored since it is not accurate.

Workaround: It is possible for BIOS to workaround this issue by masking off bit 0 in the EAX register where BIST results are written.

Status: For the steppings effected, see the *Summary Table of Changes*.

O49 False data strobe glitch machine check error may occur when the machine check handler is enabled

Problem: When the machine check handler is enabled, a false data strobe glitch error may occur and be logged into the MC0_Status register. In this case the MC0_STATUS register will contain 0xA20000001040080F. Bit 22 being set indicates that a strobe glitch error has occurred.

Implication: Data strobe glitch machine check error may occur when the machine check handler is enabled even though no actual data strobe glitch has occurred.

Workaround: None at this time.

Status: For the steppings effected, see the *Summary Table of Changes*.

O50 Processor may hang under certain frequencies and 12.5% STPCLK# duty cycle

Problem: If a system de-asserts STPCLK# at a 12.5% duty cycle, the processor is running below 2 GHz, and the processor thermal control circuit (TCC) on-demand clock modulation is active, the processor may hang. This erratum does not occur under the automatic mode of the TCC.

Implication: When this erratum occurs, the processor will hang.

Workaround: If use of the on-demand mode of the processor's TCC is desired in conjunction with STPCLK# modulation, then assure that STPCLK# is not asserted at a 12.5% duty cycle.

Status: For the steppings effected, see the *Summary Table of Changes*.

O51 BPM[5:3]# V_{IL} does not meet specification

Problem: The V_{IL} for BPM[5:3]# is specified as 0.9 * GTLREF [V]. Due to this erratum the V_{IL} for these signals is 0.9 * GTLREF -.100 [V].

Implication: The processor requires a lower input voltage than specified to recognize a low voltage on the BPM[5:3]# signals.

Workaround: When intending to drive the BPM[5:3]# signals low, ensure that the system provides a voltage lower than 0.9 * GTLREF -.100 [V].

Status: For the steppings effected, see the *Summary Table of Changes*.

O52 STPCLK# signal assertion under certain conditions may cause a system hang

Problem: The assertion of STPCLK# signal before a logical processor awakens from the “Wait-for-SIPI” state for the first time, may cause a system hang. A processor supporting HT Technology may fail to initialize appropriately, and may not issue a Stop Grant Acknowledge special bus cycle in response to the second STPCLK# assertion.

Implication: When this erratum occurs in an HT Technology enabled system, it may cause a system hang.

Workaround: BIOS should initialize the second thread of the processor supporting HT Technology prior to STPCLK# assertion. Additionally, it is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Table of Changes*.

O53 Parity error in the L1 cache may cause the processor to hang

Problem: If a locked operation accesses a line in the L1 cache that has a parity error, it is possible that the processor may hang while trying to evict the line.

Implication: If this erratum occurs, it may result in a system hang. Intel has not observed this erratum with any commercially available software.

Workaround: None at this time.

Status: For the steppings effected, see the *Summary Table of Changes*.

O54 The TCK input in the test access port (TAP) is sensitive to low clock edge rates and prone to noise coupling onto TCK's rising or falling edges

Problem: TCK is susceptible to double clocking when low amplitude noise occurs on TCK edge, while it is crossing the receiver's transition region. TAP failures tend to increase with increases in background system noise.

Implication: This only impacts JTAG/TAP accesses to the processor. Other bus accesses are not affected.

Workaround: To minimize the effects of this issue, reduce noise on the TCK-net at the processor relative to ground, and position TCK relative to BCLK to minimize the TAP error rate. Decreasing rise times to under 800ps reduced the failure rate but does not stop all failures.

Status: For the steppings effected, see the *Summary Table of Changes*.

O55 Disabling a local APIC disables both logical processor APICs on a Hyper-Threading Technology enabled processor

Problem: Disabling a local APIC on one logical processor of a HT Technology enabled processor by clearing bit 11 of the IA32_APIC_BASE MSR will effectively disable the Local APIC on the other logical processor.

Implication: Disabling a local APIC on one logical processor prevents the other logical processor from sending or receiving interrupts. Multiprocessor Specification compliant BIOSs and multiprocessor operating systems typically leave all local APICs enabled preventing any end-user visible impact from this erratum.

Workaround: Do not disable the local APICs in a HT Technology enabled processor.

Status: For the steppings effected, see the *Summary Table of Changes*.

O56 Using STPCLK and executing code from very slow memory could lead to a system hang

Problem: The system may hang when the following conditions are met:

1. Periodic STPCLK mechanism is enabled via the chipset.
2. HT Technology is enabled.
3. One logical processor is waiting for an event (i.e. hardware interrupt).
4. The other logical processor executes code from very slow memory such that every code fetch is deferred long enough for the STPCLK to be re-asserted.

Implication: If this erratum occurs, the processor will go into and out of the sleep state without making forward progress, since the logical processor will not be able to service any pending event. This erratum has not been observed in any commercial platform running commercial software.

Workaround: None at this time.

Status: For the steppings effected, see the *Summary Table of Changes*.

O57 Simultaneous cache line eviction from L2 and L3 caches may result in the write back of stale data

Problem: If a cache line is evicted simultaneously from both the L2 and L3 caches, and the internal bus queues are full, an older L3 eviction may be allowed to remain in an internal queue entry. If in a narrow timing window an external snoop is generated the data from the older eviction may be used to respond to the external snoop.

Implication: In the event that this erratum occurs the contents of memory will be incorrect. This may result in application, operating system or system failure.

Workaround: BIOS may contain a workaround for this erratum.

Status: For the steppings effected, see the *Summary Table of Changes*.

O58 The state of the resume flag (RF flag) in a task-state segment (TSS) may be incorrect

Problem: ITP will not continue in single step execution after the first software breakpoint. ITP is unable to reset the resume flag (RF) bit in the EFLAGS Register.

Implication: The processor will break at the instruction breakpoint address instead of single stepping.

Workaround: Execution after the break will continue if DR7 bit 1 (Global Breakpoint Enable) is manually cleared.

Status: For the steppings affected, see the *Summary Table of Changes*.

O59 Changes to CR3 register do not fence pending instruction page

Problem: When software writes to the CR3 register, it is expected that all previous/outstanding code, data accesses and page walks are completed using the previous value in CR3 register. Due to this erratum, it is possible that a pending instruction page walk is still in progress, resulting in an access (to the PDE portion of the page table) that may be directed to an incorrect memory address.

Implication: The results of the access to the PDE will not be consumed by the processor so the return of incorrect data is benign. However, the system may hang if the access to the PDE does not complete with data (e.g., infinite number of retries).

Workaround: It is possible for the BIOS to have a workaround for this erratum.

Status: For the steppings affected, see the *Summary Table of Changes*.

O60 Simultaneous page-faults at similar page offsets on both logical processors of an Hyper-Threading Technology enabled processor may cause application failure

Problem: An incorrect value of CR2 may be presented to one of the logical processors of an HT Technology enabled processor if a page access fault is encountered on one logical processor in the same clock cycle that the other logical processor also encounters a page-fault. Both accesses must cross the same 4 byte aligned offset for this erratum to occur. Only a small percentage of such simultaneous accesses are vulnerable. The vulnerability of the alignment for any given fault is dependent on the state of other circuitry in the processor. Additionally, a third fault from an access that occurs sequentially after one of these simultaneous faults has to be pending at the time of the simultaneous faults. This erratum is caused by a one-cycle hole in the logic that controls the timing by which a logical processor is allowed to access an internal asynchronous fault address register. The end result is that the value of CR2 presented to one logical processor may be corrupted.

Implication: The operating system is likely to terminate the application that generated an incorrect value of CR2.

Workaround: An operating system or page management software can significantly reduce the already small possibility of encountering this failure by restarting or retrying the faulting instruction and only terminate the application on a subsequent failures of the same instruction. It is possible for BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the Summary Tables of Changes.

O61 A 16-bit address wrap resulting from a near branch (jump or call) may cause an incorrect address to be reported to the #GP exception handler

Problem: If a 16-bit application executes a branch instruction that causes an address wrap to a target address outside of the code segment, the address of the branch instruction should be provided to the general protection exception handler. It is possible that, as a result of this erratum, that the general protection handler may be called with the address of the branch target.

Implication: A 16-bit software environment that is effected by this erratum will see that the address reported by the exception handler points to the target of the branch rather than the address of the branch instruction.

Workaround: None at this time.

Status: For the steppings effected, see the *Summary Table of Changes*.

O62 Incorrect PIROM L3 cache present value

Problem: The L3 Cache Present bit, located at 78:0h in the processor information read only memory (PIROM), should be programmed to 3Fh - indicating that an L3 cache is present. The L3 cache present bit value returned is 3Eh, which indicates that the L3 cache is not present. L3 cache size is not affected by the L3 cache present bit. L3 cache size can be determined by reading the L3 cache size register located at 29 -2Ah in the PIROM.

Implication: Intel expects the impact of this issue to be limited to incorrect L3 cache identification on BIOS POST or OS system information screens. L3 cache functionality is not affected by this incorrect value. However, BIOS developers and system board manufacturers should judge the impact of this L3 cache issue on their existing platform designs.

Workaround: None at this time.

Status: For the steppings affected, see the Summary Tables of Changes.

O63 Locks and SMC detection may cause the processor to temporarily hang

Problem: The processor may temporarily hang in an HT Technology enabled system, if one logical processor executes a synchronization loop that includes one or more bus locks and is waiting for release by the other logical processor. If the releasing logical processor is executing instructions that are within the detection range of the self modifying code (SMC) logic, then the processor may be locked in the synchronization loop until the arrival of an interrupt or other event.

Implication: If this erratum occurs in an HT Technology enabled system, the application may temporarily stop making forward progress. Intel has not observed this erratum with any commercially available software.

Workaround: None at this time.

Status: For the steppings affected, see the Summary of Table of Changes.

O64 Incorrect debug exception (#DB) may occur when a data breakpoint is set on an FP instruction

Problem: The default Microcode Floating Point Event Handler routine executes a series of loads to obtain data about the FP instruction that is causing the FP event. If a data breakpoint is set on the instruction causing the FP event, the load in the microcode routine will trigger the data breakpoint resulting in a Debug Exception.

Implication: An incorrect debug exception (#DB) may occur if data breakpoint is placed on an FP instruction. Intel has not observed this erratum with any commercially available software or system.

Workaround: None at this time.

Status: For the steppings affected, see the Summary of Table of Changes.

O65 Modified cache line eviction from L2 cache may result in write back of stale data

Problem: It is possible for a modified cache line to be evicted from the L2 cache just prior to another update to the same line by software. In rare circumstances, the processor may accrue two bus queue entries that have the same address but have different data. If an external snoop is generated in a narrow timing window, the data from the older eviction may be used to respond to the external snoop.

Implication: In the event that this erratum occurs, the contents of memory will be incorrect. This may result in application, operating system, or system failure.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the Summary of Table of Changes.

O66 xAPIC may not report some illegal vector errors

Problem: The local xAPIC has an error status register, which records all errors. The bit 6 (the Receive illegal Vector bit) of this register, is set when the local xAPIC detects an illegal vector in a received message. When an illegal vector error is received on the same internal clock that the error status register is being written (due to a previous error), bit 6 does not get set and illegal vector errors are not flagged.

Implication: The xAPIC may not report some Illegal Vector errors when they occur at approximately the same time as other xAPIC errors. The other xAPIC errors will continue to be reported.

Workaround: None at this time.

Status: For the steppings affected, see the Summary of Table of Changes.

O67 Incorrect duty cycle is chosen when On-Demand Clock Modulation is enabled in a processor supporting Hyper-Threading Technology

Problem: When a processor supporting HT Technology enables On-Demand Clock modulation on both logical processors, the processor is expected to select the lowest duty cycle of the two potentially different values. When one logical processor enters the AUTOHALT state, the duty cycle implemented should be unaffected by the halted logical processor. Due to this erratum, the duty cycle is incorrectly chosen to be the higher duty cycle of both logical processors.

Implication: Due to this erratum, higher duty cycle may be chosen when the On-Demand Clock modulation is enabled on both logical processors.

Workaround: None at this time.

Status: For the steppings affected, see the Summary of Table of Changes.

O68 Memory aliasing of pages as uncacheable memory type and write back (WB) may hang the system

Problem: When a page is being accessed as either UC or WC and WB, under certain bus and memory timing conditions, the system may **loop** in a continual sequence of UC fetch, implicit WB, and Request RFO retries.

Implication: This erratum has not been observed in any commercially available operating system or application. The aliasing of memory regions, a condition necessary for this erratum to occur, is documented as being unsupported in the *IA-32 Intel® Architecture Software Developer's Manual*, Volume 3, Section 10.12.4 However, if this erratum occurs the system may hang.

Workaround: The pages should not be mapped as either UC or WC and WB at the same time.

Status: For the steppings affected, see the Summary of Table of Changes.

O69 A timing marginality in the Instruction Decoder unit may cause an unpredictable application behavior and/or system hang

Problem: A timing marginality may exist in the clocking of the instruction decoder unit which leads to a circuit slowdown in the read path from the Instruction Decode PLA circuit. This timing marginality may not be visible for some period of time.

Implication: When this erratum occurs, an incorrect instruction stream may be executed resulting in an unpredictable application behavior and/or system hang.

Workaround: It is possible for the BIOS to contain a workaround for this erratum. BIOS must load the microcode update during the BIOS POST time prior to memory initialization.

Status: For the steppings affected, see the Summary of Table of Changes.

O70 Missing Stop Grant Acknowledge special bus cycle may cause a system hang

Problem: If a Stop Grant Acknowledge special bus cycle is deferred by the processor for a period of time long enough for the chipset to de-assert and then re-assert STPCLK# signal, a processor supporting Hyper-Threading Technology may fail to detect the de-assertion and re-assertion of STPCLK# signal. When this occurs, the processor will not issue a Stop Grant Acknowledge special bus cycle in response to the second STPCLK# assertion.

Implication: When this erratum occurs in an Hyper-Threading Technology enabled system, it may cause a system hang.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Table of Changes*.

O71 Machine check exceptions may not update Last-Exception Record MSRs (LERs)

Problem: The Last-Exception Record MSRs (LERs) may not get updated when machine check exceptions (MCE) occur.

Implication: When this erratum occurs, the LER may not contain information relating to the MCE. They will contain information relating to the exception prior to the MCE.

Workaround: None at this time.

Status: For the steppings affected, see the *Summary Table of Changes*.

O72 Stores to page tables may not be visible to page walks for subsequent loads without serializing or invalidating the page table entry

Problem: Under rare timing circumstances, a page table load on behalf of a programmatically younger memory access may not get data from a programmatically older store to the page table entry if there is not a fencing operation or page translation invalidate operation between the store and the younger memory access. Refer to the *IA-32 Intel® Architecture Software Developer's Manual* for the correct way to update page tables. Software that conforms to the *IA-32 Intel® Architecture Software Developer's Manual* will operate correctly.

Implication: If the guidelines in the *IA-32 Intel® Architecture Software Developer's Manual* are not followed, stale data may be loaded into the processor's translation lookaside buffer (TLB) and used for memory operations. This erratum has not been observed with any commercially available software.

Workaround: The guidelines in the *IA-32 Intel® Architecture Software Developer's Manual* should be followed.

Status: For the steppings affected, see the *Summary Table of Changes*.

O73 A timing marginality in the Arithmetic Logic Unit (ALU) may cause indeterminate behavior

Problem: A timing marginality may exist in the clocking of the ALU which leads to a slowdown in the speed of the circuit's operation. This could lead to incorrect behavior of the ALU.

Implication: When this erratum occurs, unpredictable application behavior and/or system hang may occur.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Table of Changes*.

O74 With Trap Flag (TF) asserted, FP instruction that triggers an unmasked FP exception may take single step trap before retirement of instruction

Problem: If an FP instruction generates an unmasked exception with the EFLAGS.TF=1, it is possible for external events to occur, including a transition to a lower power state. When resuming from the lower power state, it may be possible to take the single step trap before the execution of the original FP instruction completes.

Implication: A Single Step trap will be taken when not expected.

Workaround: None at this time.

Status: For the steppings affected, see the *Summary Table of Changes*.

O75 PDE/PTE loads and continuous locked updates to the same cache line may cause a system livelock

Problem: In a multiprocessor configuration, if one processor is continuously doing locked updates to a cache line that is being accessed by another processor doing a page table walk, the page table walk may not complete.

Implication: Due to this erratum, the system may livelock until some external event interrupts the locked update. Intel has not observed this erratum with any commercially available software.

Workaround: None at this time.

Status: For the steppings affected, see the *Summary Table of Changes*.

O76 Branch Trace Store (BTS) and Precise Event Based Sampling (PEBS) may update memory outside the BTS/PREBS buffer

Problem: If the BTS/PREBS buffer is defined such that:

- The difference between BTS/PREBS buffer base and BTS/PREBS absolute maximum is not an integer multiple of the corresponding record sizes.
- BTS/PREBS absolute maximum is less than a record size from the end of the virtual address space.
- The record that would cross BTS/PREBS absolute maximum will also continue past the end of the virtual address space.

A BTS/PREBS record can be written that will wrap at the 4G boundary (IA-32) or 2^64 boundary (Intel® Extended Memory 64 Technology (Intel® EM64T) mode), and write memory outside of the BTS/PREBS buffer.

Implication: Software that uses BTS/PREBS near the 4G boundary (IA-32) or 2^64 boundary (Intel EM64T mode), and defines the buffer such that it does not hold an integer multiple of records can update memory outside the BTS/PREBS buffer.

Workaround: Define BTS/PREBS buffer such that BTS/PREBS absolute maximum minus BTS/PREBS buffer base is integer multiple of the corresponding record sizes as recommended in the *IA-32 Intel® Architecture Software Developer's Manual*, Volume 3.

Status: For the steppings affected, see the *Summary Table of Changes*.

O77 Memory Ordering Failure may occur with Snoop Filtering Third-Party Agents after issuing and completing a BWIL (Bus Write Invalidate Line) or BLW (Bus Locked Write) transaction

Problem: Under limited circumstances, the processors may, after issuing and completing a BWIL or BLW transaction, retain data from the addressed cache line in shared state even though the specification

requires complete invalidation. This data retention may also occur when a BWIL transaction's self-snooping yields HITM snoop results.

Implication: A system may suffer memory ordering failures if its central agent incorporates coherence sequencing which depends on full self-invalidation of the cache line associated with (1) BWIL and BLW transactions, or (2) all HITM snoop results without regard to the transaction type and snoop results' source.

Workaround:

1. The central agent can issue a bus cycle that causes a cache line to be invalidated (Bus Read Invalidate Line (BRIL) or BWIL transaction) in response to a processor-generated BWIL (or BLW) transaction to insure complete invalidation of the associated cache line. If there are no intervening processor-originated transactions to that cache line, the central agent's invalidating snoop will get a clean snoop result.

Or

2. Snoop filtering central agents can:

   a. Not use processor-originated BWIL or BLW transactions to update their snoop filter information, or

   b. Update the associated cache line state information to shared state on the originating bus (rather than invalid state) in reaction to a BWIL or BLW.

Status: For the steppings affected, see the *Summary Table of Changes*.

O78 Control Register 2 (CR2) can be updated during a REP MOVS/STOS instruction with Fast Strings enabled

Problem: Under limited circumstances while executing a `REP MOVS/STOS` string instruction, with fast strings enabled, it is possible for the value in CR2 to be changed as a result of an interim paging event, normally invisible to the user. Any higher priority architectural event that arrives and is handled while the interim paging event is occurring may see the modified value of CR2.

Implication: The value in CR2 is correct at the time that an architectural page fault is signaled. Intel has not observed this erratum with any commercially available software.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Table of Changes*.

Specification Changes

There are no new Specification Changes for this month.

The Specification Changes listed in this section apply to the following documents:

- *Intel® Xeon™ Processor MP Datasheet* (Order Number 290740)
- *Intel® Xeon™ Processor MP with up to 4MB L3 Cache (on 0.13-Micron Process) Datasheet* (Order Number 251931)
- *IA-32 Intel® Architecture Software Developer's Manual*, Volumes 1, 2A, 2B and 3 (Order Numbers 253665, 253666, 253667, and 253668, respectively)
- *Intel® Extended Memory 64 Technology Software Developer's Guide*, Volume 1 (Order Number 300834)

  Here is the link: http://developer.intel.com/technology/64bitextensions/300834.htm
- *Intel® Extended Memory 64 Technology Software Developer's Guide*, Volume 2 (Order Number 300835)

  Here is the link: http://developer.intel.com/technology/64bitextensions/300835.htm

All Specification Changes will be incorporated into a future version of the appropriate Intel Xeon processor documentation.

Specification Clarifications

The Specification Clarifications listed in this section apply to the following documents:

- *Intel® Xeon™ Processor MP Datasheet* (Order Number 290740)
- *Intel® Xeon™ Processor MP with up to 4MB L3 Cache (on 0.13-Micron Process) Datasheet* (Order Number 251931)
- *IA-32 Intel® Architecture Software Developer's Manual*, Volumes 1, 2A, 2B and 3 (Order Numbers 253665, 253666, 253667, and 253668, respectively)
- *Intel® Extended Memory 64 Technology Software Developer's Guide*, Volume 1 (Order Number 300834)

  Here is the link: http://developer.intel.com/technology/64bitextensions/300834.htm
- *Intel® Extended Memory 64 Technology Software Developer's Guide*, Volume 2 (Order Number 300835)

  Here is the link: http://developer.intel.com/technology/64bitextensions/300835.htm

All Specification Clarifications will be incorporated into a future version of the appropriate Intel Xeon processor documentation.

O1 Specification Clarification with respect to Time-Stamp Counter

In the "Debugging and Performance Monitoring" chapter (Section 15.8, Section 15.10.9 and Section 15.10.9.3) of the *IA-32 Intel® Architecture Software Developer's Manual, Volume 3: System Programming Guide*, the Time-Stamp Counter definition has been updated to include support for the future processors. This change will be incorporated in the next revision of the *IA-32 Intel® Architecture Software Developer's Manual*.

15.8 Time-Stamp Counter

The IA-32 architecture (beginning with the Pentium® processor) defines a time-stamp counter mechanism that can be used to monitor and identify the relative time occurrence of processor events. The counter's architecture includes the following components:

- **TSC flag** — A feature bit that indicates the availability of the time-stamp counter. The counter is available in an IA-32 processor implementation if the function CPUID.1:EDX.TSC[bit 4] = 1.
- **IA32_TIME_STAMP_COUNTER MSR** (called TSC MSR in P6 family and Pentium processors) — The MSR used as the counter.
- **RDTSC instruction** — An instruction used to read the time-stamp counter.
- **TSD flag** — A control register flag is used to enable or disable the time-stamp counter (enabled if CR4.TSD[bit 2] = 1).

The time-stamp counter (as implemented in the P6 family, Pentium, Pentium M, Pentium 4, and Intel® Xeon™ processors) is a 64-bit counter that is set to 0 following a RESET of the processor. Following a RESET, the counter will increment even when the processor is halted by the HLT instruction or the external STPCLK# pin. Note that the assertion of the external DPSLP# pin may cause the time-stamp counter to stop.

Members of the processor families increment the time-stamp counter differently:

- For Pentium M processors (family [06H], models [09H, 0DH]); for Pentium 4 processors, Intel Xeon processors (family [0FH], models [00H, 01H, or 02H]); and for P6 family processors: the time-stamp counter increments with every internal processor clock cycle. The internal processor clock cycle is determined by the current core-clock to bus-clock ratio. Intel SpeedStep® technology transitions may also impact the processor clock.
- For Pentium 4 processors, Intel Xeon processors (family [0FH], models [03H and higher]): the time-stamp counter increments at a constant rate. That rate may be set by the maximum core-clock to bus-clock ratio of the processor or may be set by the frequency at which the processor is booted. The specific processor configuration determines the behavior. Constant TSC behavior ensures that the duration of each clock tick is uniform and supports the use of the TSC as a wall clock timer even if the processor core changes frequency. This is the architectural behavior moving forward.

Note: To determine average processor clock frequency, Intel recommends the use of Performance Monitoring logic to count processor core clocks over the period of time for which the average is required. See Section 15.10.9 and Appendix A in this manual for more information.

The RDTSC instruction reads the time-stamp counter and is guaranteed to return a monotonically increasing unique value whenever executed, except for a 64-bit counter wraparound. Intel guarantees that the time-stamp counter will not wraparound within 10 years after being reset. The period for counter wrap is longer for Pentium 4, Intel Xeon, P6 family, and Pentium processors.

Normally, the RDTSC instruction can be executed by programs and procedures running at any privilege level and in virtual-8086 mode. The TSD flag allows use of this instruction to be restricted to programs and procedures running at privilege level 0. A secure operating system would set the TSD flag during system initialization to disable user access to the time-stamp counter. An operating system that disables user access to the time-stamp counter should emulate the instruction through a user-accessible programming interface.

The RDTSC instruction is not serializing or ordered with other instructions. It does not necessarily wait until all previous instructions have been executed before reading the counter. Similarly, subsequent instructions may begin execution before the RDTSC instruction operation is performed.

The RDMSR and WRMSR instructions read and write the time-stamp counter, treating the time-stamp counter as an ordinary MSR (address 10H). In the Pentium 4, Intel Xeon, and P6 family processors, all 64-bits of the time-stamp counter are read using RDMSR (just as with RDTSC). When WRMSR is used to write the time-stamp counter on processors before family [0FH], models [03H, 04H]: only the low order 32-bits of the time-stamp counter can be written (the high-order 32 bits are cleared to 0). For family [0FH], models [03H, 04H]: all 64 bits are writeable.

15.10.9 Counting Clocks

The count of cycles, also known as clockticks, forms a the basis for measuring how long a program takes to execute. Clockticks are also used as part of efficiency ratios like cycles per instruction (CPI). Processor clocks may stop ticking under circumstances like the following:

- The processor is halted when there is nothing for the CPU to do. For example, the processor may halt to save power while the computer is servicing an I/O request. When Hyper-Threading Technology is enabled, both logical processors must be halted for performance-monitoring counters to be powered down.
- The processor is asleep as a result of being halted or because of a power-management scheme. There are different levels of sleep. In the some deep sleep levels, the time-stamp counter stops counting.

There are three ways to count processor clock cycles to monitor performance. These are:

- **Non-halted clockticks** — Measures clock cycles in which the specified logical processor is not halted and is not in any power-saving state. When Hyper-Threading Technology is enabled, this these ticks can be measured on a per-logical-processor basis.
- **Non-sleep clockticks** — Measures clock cycles in which the specified physical processor is not in a sleep mode or in a power-saving state. These ticks cannot be measured on a logical-processor basis.
- **Time-stamp counter** — Some processor models permit clock cycles to be measured when the physical processor is not in deep sleep (by using the time-stamp counter and the RDTSC instruction). Note that such ticks cannot be measured on a per-logical-processor basis. See Section 10.8 for detail on processor capabilities.

The first two methods use performance counters and can be set up to cause an interrupt upon overflow (for sampling). They may also be useful where it is easier for a tool to read a performance counter than to use a time stamp counter (the timestamp counter is accessed using the RDTSC instruction).

For applications with a significant amount of I/O, there are two ratios of interest:

- **Non-halted CPI** — Non-halted clockticks/instructions retired measures the CPI for phases where the CPU was being used. This ratio can be measured on a logical-processor basis when Hyper-Threading Technology is enabled.
- **Nominal CPI** — Time-stamp counter ticks/instructions retired measures the CPI over the duration of a program, including those periods when the machine halts while waiting for I/O.

15.10.9.3 Incrementing the Time-Stamp Counter

The time-stamp counter increments when the clock signal on the system bus is active and when the sleep pin is not asserted. The counter value can be read with the RDTSC instruction.

The time-stamp counter and the non-sleep clockticks count may not agree in all cases and for all processors. See Section 10.8 for more information on counter operation.

Documentation Changes

There are no new Documentation Changes for this month.

The Documentation Changes listed in this section apply to the following documents:

- *Intel® Xeon™ Processor MP Datasheet* (Order Number 290740)
- *Intel® Xeon™ Processor MP with up to 4MB L3 Cache (on 0.13-Micron Process) Datasheet* (Order Number 251931)
- *IA-32 Intel® Architecture Software Developer's Manual*, Volumes 1, 2A, 2B and 3 (Order Numbers 253665, 253666, 253667, and 253668, respectively)
- *Intel® Extended Memory 64 Technology Software Developer's Guide*, Volume 1 (Order Number 300834)

  Here is the link: http://developer.intel.com/technology/64bitextensions/300834.htm
- *Intel® Extended Memory 64 Technology Software Developer's Guide*, Volume 2 (Order Number 300835)

  Here is the link: http://developer.intel.com/technology/64bitextensions/300835.htm

All Documentation Changes will be incorporated into a future version of the appropriate Intel Xeon processor documentation.